1

What Is Windows PowerShell?

COPYRIGHTED MATERIAL

Windows PowerShell is the extensible command-line interface shell and scripting language that provides a command-line environment for interactive exploration and administration of computers. In addition, it provides developers with an opportunity to script these commands, enabling them to be automated, scheduled, and run multiple times.

This chapter covers the following topics:

- A brief overview of command-line interfaces
- Prerequisites for installing PowerShell
- Installing PowerShell 2.0 CTP3.

Command-Line Interfaces versus Graphical User Interfaces

Before UNIX, Linux, and Windows surfaced in the information technology market, input media such as punched card and punched tape were used. All the input and instructions to the computer used command lines.

When UNIX and Linux were released, administrators started managing the operating system using command-line interfaces, which were also used by the day-to-day users and programmers to interact with the operating system.

UNIX and Linux are built on command-line interfaces, so there has always been administrative scripting and different shells, such as the Korn shell, the C shell, and the Bourne shell. Programming languages such as TCL and PERL were immediately available.

Originally, when Microsoft released MS-DOS, it was not used as a shell. It was entirely a standalone operating system. The initial and original Microsoft Windows release was a graphical shell that sat on top of the MS-DOS operating system. Once Windows NT was introduced, this situation was reversed. MS-DOS became the shell in the graphical Windows operating system.

Graphical user interfaces (GUIs) were basically developed for users with less technical background who were looking for a friendly interface. Because graphical interfaces are limited to fewer functions, once you hit their limitations you will start relying on the command-line interface. For example, with a GUI, if you want to rename all the extensions of a group of files and suffix each file with the current day's timestamp, it will take a while because you have to select and rename each file individually.

Therefore, command-line interfaces and their commands are very commonly used for administrative tasks and automation. They also help to consolidate functionality in batches, through MS-DOS batch files.

`command.com` was used as the command-line user interface in early versions of Microsoft Windows. `cmd.exe` was introduced in Windows NT.

When administrators reached the limit of command-line batch files, they started looking for a language that could do both command shell functions and programming. Microsoft introduced the Visual Basic scripting language, which helped administrators initially. However, there were limitations in VBScript as well. Administrators started relying on Windows Management Instrumentation (WMI) and COM objects introduced later by Microsoft for many other administrative functions.

The administrative command-line interface started becoming more complicated with internal DOS commands, external DOS commands, batch files, executables, VBScripts, WMI, and so forth.

It was with all of this in mind that Microsoft developed and introduced Windows PowerShell.

Prerequisites for Installing Windows PowerShell 2.0

Windows PowerShell can be installed and run on Windows XP, Windows Vista, Windows Server 2003, and Windows Server 2008. Although Windows PowerShell is included as part of Windows Server 2008, it is not installed by default on Windows XP, Windows 2003, or Windows Vista. At the time of writing, Windows PowerShell 1.0 is visible as a feature in Windows Server 2008 that can be turned on.

Windows PowerShell 1.0 is also installed with Exchange Server 2007, System Center Operations Manager 2007, System Center Data Protection Manager V2, and System Center Virtual Machine Manager because they leverage Windows PowerShell to improve administrator control, efficiency, and productivity.

This book uses Windows PowerShell 2.0 CTP3. Before installing Windows PowerShell 2.0 CTP3, you should ensure that your system has the following software programs it requires:

- Windows XP Service Pack 3, Windows 2003 Service Pack 2, Windows Vista Service Pack 1, or Windows Server 2008
- Microsoft .NET Framework 2.0 or greater

- Windows Remote Management 2.0 CTP3 for Windows PowerShell remoting and background jobs
- Microsoft .NET Framework 3.5 Service Pack 1 for Windows PowerShell Integrated Scripting Environment (ISE) and the Out-GridView cmdlet

If .NET Framework 2.0 is not installed on your computer, the error message shown in Figure 1-1 will pop up when you try to install Windows PowerShell 2.0 CTP3.

Windows PowerShell Setup Wizard
Cannot install Windows PowerShell(TM) V2 (CTP3) because it requires an updated version of the .NET Framework. Please install .NET Framework 2.0 RTM from http://go.microsoft.com/fwlink/?LinkID=100351&clcid=0x409 and rerun Windows PowerShell(TM) V2 (CTP3) Setup
OK

Figure 1-1

After installing the .NET Framework, if you continue to install on a non-standard operating system such as Windows Server 2000, Windows NT, and so on, you may get the error shown in Figure 1-2.

Figure 1-2

Although Figure 1-2 shows v1.0 in the title bar and file path, it is actually PowerShell 2.0 CTP3. Community Technology Preview, also known as CTP3, is basically the bug fix that Microsoft relies on, adding new features based on feedback from the technology community. The folder may be changed to v2.0 once the release to manufacturing (RTM) *version is released.*

In order to check the prerequisites for the PowerShell installation, we created a batch file to run all the necessary checks. Listing 1-1 shows the MS-DOS batch script, `CheckPowershellPreqs.bat`, which uses Windows Management Instrumentation Command-line (WMIC) to check the requirements before you install Windows PowerShell. WMIC enables you to access WMI information via the command line. In some respects, you can think of WMIC as an early prototype of PowerShell. However, WMIC can only output its results as text. It doesn't return programming objects that can be further processed, as Window PowerShell does, and as you will see later.

Listing 1-1: CheckPowershellPreqs.bat

```
@ECHO Off

REM ****************************************************************
REM *Objective: TO check if the current windows version is          *
REM *           Compatible for PowerShell 2.0 and its pre-requisites*
REM *Created by: Yan and MAK                                       *
REM *Created Date: 2008/09/01                                      *
```

Continued

Listing 1-1: CheckPowershellPreqs.bat *(continued)*

```
REM *                                                                 *
REM *******************************************************************


SET OS_VERSION=
SET Service_Pack=

REM Find OS version
FOR /F "delims== tokens=2" %%i IN ('wmic os get Version /value')
   DO SET OS_VERSION=%%i

IF NOT DEFINED OS_VERSION (
    ECHO WMIC is not installed on this system.
    GOTO :END_SCRIPT
)

REM Find service pack value
FOR /F "delims== tokens=2" %%i IN ('wmic os get ServicePackMajorVersion /value')
   DO SET Service_Pack=%%i

REM Windows XP
IF "%OS_VERSION%"=="5.1.2600" (
    @IF "%Service_Pack%" LSS "3" (
       ECHO %OS_NAME% Service Pack 3 is required
       GOTO :END_SCRIPT
    )
    GOTO :DOTNETFRAMEWORK_CHECK
)

REM Windows Server 2003
IF "%OS_VERSION%"=="5.2.3790" (
    @IF "%Service_Pack%" LSS "2" (
       ECHO %OS_NAME% Service Pack 2 is required
       GOTO :END_SCRIPT
    )
    GOTO :DOTNETFRAMEWORK_CHECK
)


REM Windows Vista
IF "%OS_VERSION%"=="6.0.6001" (
    @IF "%Service_Pack%" LSS "1" (
       ECHO %OS_NAME% Service Pack 1 is required
       GOTO :END_SCRIPT
    )
    GOTO :DOTNETFRAMEWORK_CHECK
)

IF "%OS_VERSION%" GTR "5.2.3790" (
    GOTO :DOTNETFRAMEWORK_CHECK
)
ELSE (
    GOTO :END_SCRIPT
)
```

Listing 1-1: CheckPowershellPreqs.bat *(continued)*

```
REM Check .NET framework
:DOTNETFRAMEWORK_CHECK
wmic product where (caption like "Microsoft .NET Framework%%") get Version /value |
   findstr "=[2-9]\.*" > nul
IF "%ERRORLEVEL%"=="1" (
    ECHO .NET Framework 2.0 or greater is required
    GOTO :END_SCRIPT
)

REM Check Windows remote management
:WINRMCHECK
wmic path Win32_service where caption="Windows Remote Management (WS-Management)" |
   findstr /i "(WS-Management)" > nul
IF "%ERRORLEVEL%"==1 (
    ECHO Windows Remote Management is required
    GOTO :END_SCRIPT
)

ECHO Your system meets the requirements

:END_SCRIPT
```

From a Windows command console, run `CheckPowershellPreqs.bat` from the script directory `C:\DBAScripts`. You should get a message similar to the one shown in Figure 1-3 if your system meets the requirements for a PowerShell installation.

Figure 1-3

You may run into an error as follows on Windows Server 2003:

```
ERROR:
Code = 0x80041010
Description = Invalid class
Facility = WMI
```

This means that you need to install the WMI Windows Installer Provider in order for WMIC to work. To do so, open Control Panel ➢ Add/Remove Programs. Select Add/Remove Windows Components. Double-click Management and Monitoring Tools. Select WMI Windows Installer Provider, and then click OK to install.

If your operating system is Windows XP Service Pack 2 and you install Windows PowerShell 2.0 CTP3, it may or may not work. The officially supported service pack for Windows XP when installing Windows

PowerShell 2.0 CPT3 is Service Pack 3, so when executing `C:\DBAScripts\CheckPowershellPreqs.bat` it will complain that Service Pack 3 is required.

Installing the Microsoft .NET Framework

Windows PowerShell integrates with the .NET Framework and provides a shell environment to perform administrative tasks. PowerShell exposes the .NET classes as built-in commands, and when these commands are executed they produce one or more structured objects as output.

Download the .NET Framework from `http://download.microsoft.com` and install .NET Framework 2 or later. It is recommended that you use .NET 3.5, although all the code in this book works on .NET Framework 2 and later. In Chapter 11, you will see that an SQL Server 2008 installation installs .NET 3.5 by default.

Installing Windows Remote Management

Another prerequisite is to install Windows Remote Management (WinRM). WinRM is required for Windows PowerShell remoting and background jobs. You can download WinRM 2.0 CTP3 (also known as Ws-Management) from `https://connect.microsoft.com/WSMAN/Downloads` for Windows Vista and Windows Server 2008. The installation procedures of the executable `Windows6.0-KB950099-x86.msu` for 32-bit and `Windows6.0-KB950099-x64.msu` for 64-bit are shown in Figure 1-4 and Figure 1-5.

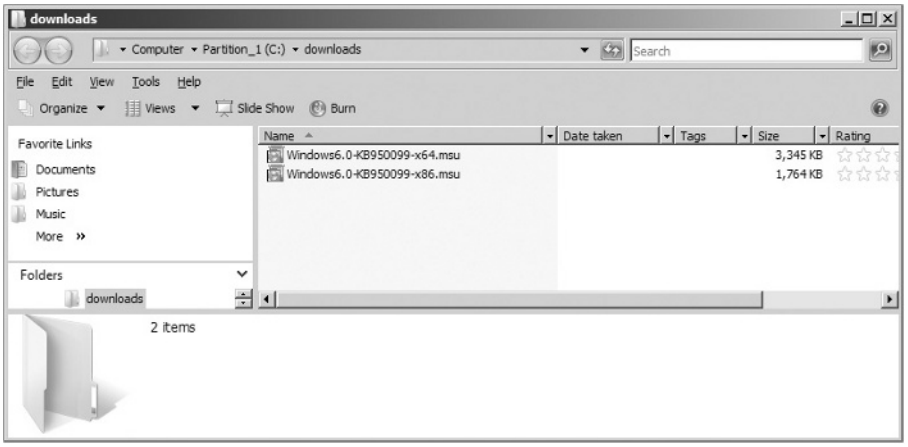

Figure 1-4

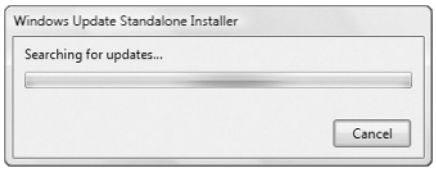

Figure 1-5

Double-click on `Windows6.0-KB950099-x86.msu` if you are installing WinRM on a 32-bit Windows operating system. Double-click on `Windows6.0-KB950099-x64.msu` if you are installing it on a 64-bit Windows operating system.

If you already have Windows PowerShell 1.0 installed on your machine, then you have to uninstall PowerShell 1.0 first and then install PowerShell 2.0 CTP3.

To uninstall Windows PowerShell 1.0, follow these steps:

1. Click Start ➪ Run, type **appwiz.cpl**, and then click OK.
2. Select the Show Updates check box (on the top in the middle).
3. In the list of currently installed programs, click Windows PowerShell(TM) 1.0, and then click Remove. If you don't see Windows PowerShell(TM) 1.0, please look for Windows Hotfix 926139, 926140, or 926141.
4. Follow the instructions to remove Windows PowerShell(TM) 1.0.

If you already have Windows PowerShell 2.0 CTP1 or CTP2, you have to uninstall them also. They appear as ''Windows PowerShell V2 (TM)'' in the program list.

In Windows 2008, Windows PowerShell 1.0 is made visible as a Windows feature. You can disable and uninstall the `PowerShell` feature as described here:

1. Click Start at the Windows taskbar.
2. Click Server Manager ➢ Features ➢ Add Feature, and uncheck the Windows PowerShell check box.
3. Click Next to uninstall.

Now that you have the acceptable operating system, the .NET Framework, and Windows Remote Management, the next section provides the steps needed to actually install Windows PowerShell 2.0 CTP3.

Installing Windows PowerShell

To install Windows PowerShell on Windows XP or Windows 2003 systems, do the following:

1. Download the Windows PowerShell 2.0 CTP3 installation file from `www.Microsoft.com/downloads`. The name of the installation file varies according to platform, operating system, and language pack. Choose the appropriate version for your operating system. If you have a 64-bit Windows operating system, then please download `PowerShell_Setup_amd64.msi`, as shown in Figure 1-6. If you have a 32-bit Windows operating system, then please download `PowerShell_Setup_x86.msi`, as shown in Figure 1-7.
2. After downloading the appropriate version for your operating system, you will see the initial screen of the installation wizard, similar to the one shown in Figure 1-8. Click Next.
3. Accept the license agreement, as shown in Figure 1-9, and click Next.

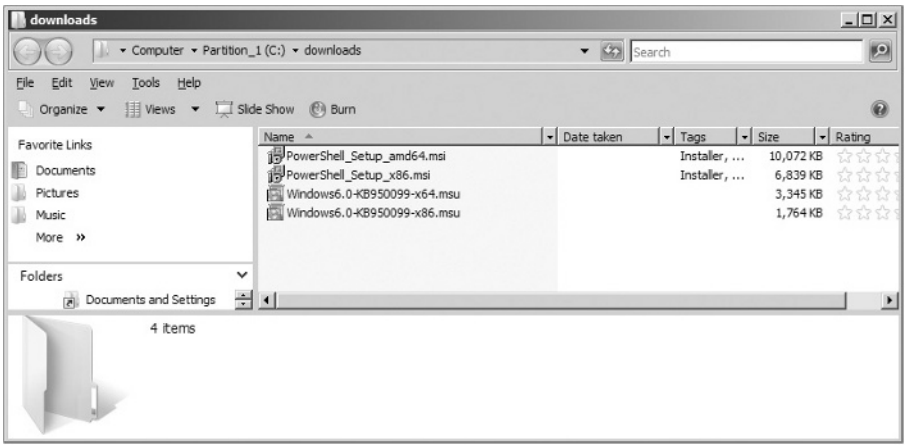

Figure 1-6

File Download - Security Warning
Do you want to run or save this file?
Name: PowerShell_Setup_x86.msi
Type: Windows Installer Package, 6.67MB
From: download.microsoft.com
Run
Save
Cancel
While files from the Internet can be useful, this file type can potentially harm your computer. If you do not trust the source, do not run or save this software. What's the risk?

Figure 1-7

Windows PowerShell Setup Wizard
Welcome to the Windows PowerShell(TM) V2 (CTP3) Setup Wizard
This wizard will install Windows PowerShell(TM) V2 (CTP3) on your computer.
To continue, click Next.
Next >
Cancel

Figure 1-8

Windows PowerShell Setup Wizard
Please read the following license agreement carefully.
MICROSOFT PRE-RELEASE SOFTWARE LICENSE TERMS
MICROSOFT WINDOWS POWERSHELL VERSION 2.0 (COMMUNITY TECHNOLOGY PREVIEWS)
These license terms are an agreement between Microsoft Corporation (or
I accept the terms in the license agreement
I do not accept the terms in the license agreement
< Back
Next >
Cancel

Figure 1-9

4. It takes a few minutes to complete the install. As shown in Figure 1-10, click Install to begin the installation process. You will see the progress of the installation, as shown in Figure 1-11.

Windows PowerShell Setup Wizard
Start Installation
The wizard is ready to begin installation.
Click Install to begin the installation or click Cancel to exit the wizard.
< Back
Install
Cancel

Figure 1-10

5. When installation is completed successfully, you will see a screen similar to the one shown in Figure 1-12. Click Finish.

On 32-bit versions of Windows, Windows PowerShell is installed by default in the `%SystemRoot%\System32\WindowsPowerShell\v1.0` directory.

On 64-bit versions of Windows, a 32-bit version of Windows PowerShell is installed in the `%SystemRoot%\SystemWow64\WindowsPowerShell\v1.0` directory and a 64-bit version of Windows PowerShell is installed in the `%SystemRoot%\System32\WindowsPowerShell\v1.0` directory.

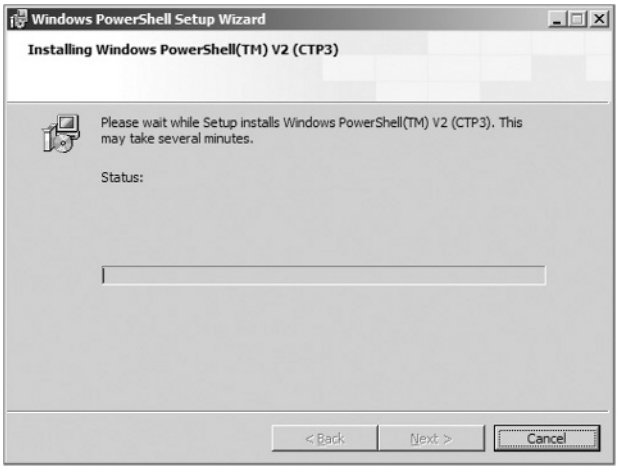
Windows PowerShell Setup Wizard
Installing Windows PowerShell(TM) V2 (CTP3)
Please wait while Setup installs Windows PowerShell(TM) V2 (CTP3). This may take several minutes.
Status:
< Back
Next >
Cancel

Figure 1-11

Windows PowerShell Setup Wizard
Completing the Windows PowerShell(TM) V2 (CTP3) Setup Wizard
You have successfully installed the Windows PowerShell(TM) V2 (CTP3).
To close this wizard, click Finish.
< Back
Finish
Cancel

Figure 1-12

This book covers Windows PowerShell 2.0 CTP3. The installation file replaces the executables in the v1.0 folder. The actual folders for Windows PowerShell may change to v2.0 when PowerShell 2.0 RTM is released.

Launching Windows PowerShell

Windows PowerShell can be launched in several ways. This section describes the different methods. No method is superior to the others. It is just a matter of preference.

Using the Command console

To launch Windows PowerShell using the command console, open the command console and then type **powershell**, as shown in Figure 1-13.

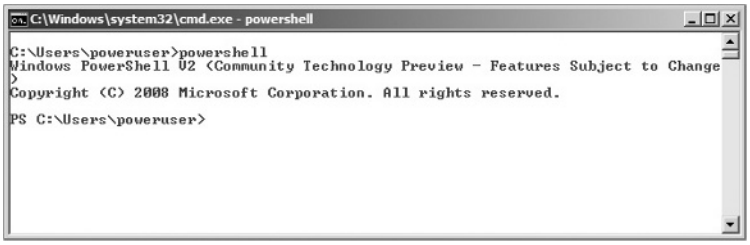
C:\Windows\system32\cmd.exe - powershell
C:\Users\poweruser>powershell
Windows PowerShell V2 (Community Technology Preview - Features Subject to Change
)
Copyright (C) 2008 Microsoft Corporation. All rights reserved.
PS C:\Users\poweruser>

Figure 1-13

Using All Programs

You can also click Start ➢ All Programs ➢ Windows PowerShell V2 (CTP3), and then select and click Windows PowerShell V2 (CTP3), as shown in Figure 1-14.

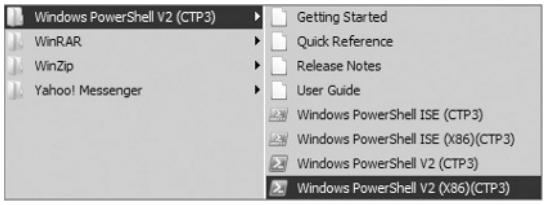

Figure 1-14

You may also see Windows PowerShell ISE (CTP3) in the Windows Program menu. The Windows PowerShell Integrated Scripting Environment (ISE) is a host application for Windows PowerShell. In Windows PowerShell ISE, you can run commands and write, test, and debug scripts in a single Windows graphical user interface. This book illustrates all Windows PowerShell-related cmdlets and scripts using a command-line interface. It does not illustrate PowerShell scripts using the ISE.

Once PowerShell is launched, you can see the command prompt. The prompt in the PowerShell command window varies according to the operating system used.

To be consistent with the PowerShell window title, you could update the shortcut, as illustrated in Figure 1-15. Right-click on the Windows PowerShell V2 (CTP3) shortcut, click Properties, and under the General tab, update the title to ''Windows PowerShell'' and click OK.

Using Start Run

You can also launch PowerShell by clicking Start ➢ Run and typing the following: **`%systemroot%\system32\windowsPowerShell\v1.0\PowerShell.exe.`** Then click OK, as shown in Figure 1-16.

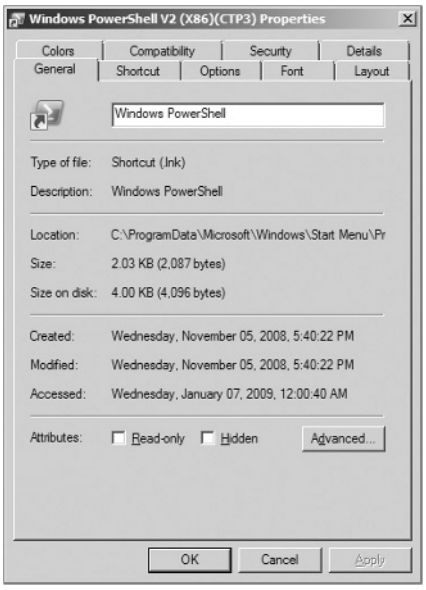

Figure 1-15

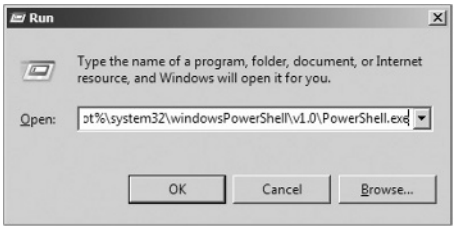

Figure 1-16

Alternatively, you can also launch PowerShell by clicking Start ➤ Run and typing **PowerShell**.

Windows finds the PowerShell.exe executable from the environment PATH variable and then launches it.

This opens the PowerShell command console, shown in Figure 1-17.

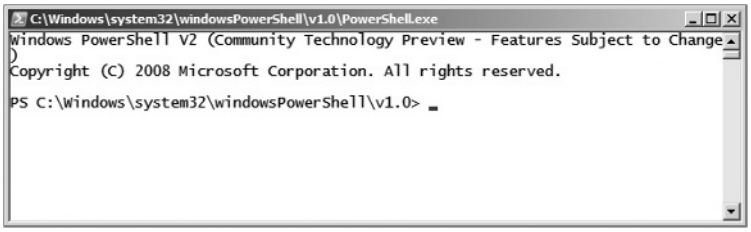
C:\Windows\system32\windowsPowerShell\v1.0\PowerShell.exe
Windows PowerShell V2 (Community Technology Preview - Features Subject to Change
)
Copyright (C) 2008 Microsoft Corporation. All rights reserved.
PS C:\Windows\system32\windowsPowerShell\v1.0>

Figure 1-17

If you are using 64-bit PowerShell, then you can launch PowerShell by clicking Start ➢ Run and typing the following: **`%systemroot%\SysWOW64\WindowsPowerShell\v1.0\PowerShell.exe.`** Then click OK, as shown in Figure 1-18.

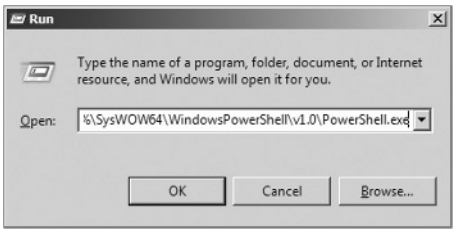

Figure 1-18

This opens the PowerShell command console, as shown in Figure 1-19.

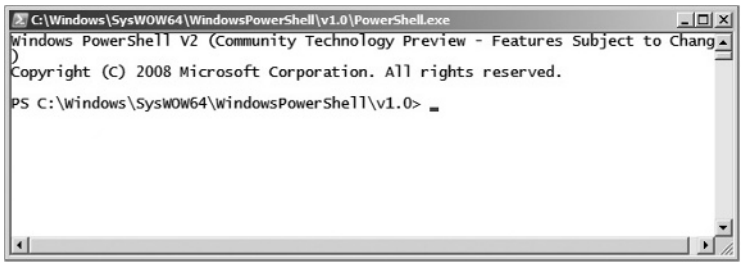
C:\Windows\SysWOW64\WindowsPowerShell\v1.0\PowerShell.exe
Windows PowerShell V2 (Community Technology Preview - Features Subject to Chang
)
Copyright (C) 2008 Microsoft Corporation. All rights reserved.
PS C:\Windows\SysWOW64\WindowsPowerShell\v1.0>

Figure 1-19

Summary

This chapter discussed the prerequisites for Windows PowerShell, described how to install Windows PowerShell 2.0 CTP3, and showed various methods to launch PowerShell.

The next chapter covers the various command types and commonly used *cmdlets* that appear throughout the book. A Windows PowerShell cmdlet is a simple command used for interaction with any managed application, including the Windows operating system and applications such as SQL Server, Microsoft Exchange, and so on.